

How to Validate Data on Maintenance View(SM30) Before Save



beyhan_meyrali
Active Contributor

2023 Mar 22 6:16 AM



13 Kudos

17,912 Views

Hi,

In this blog post, I would like to show you how to validate data on maintenance view screen before save.

Maintenance views are very useful interfaces to create a viewer and editor for database tables. And with events and search helps, we can make them even more useful.

In this blog post, we will create a table, where we can map company codes to special ranges for purchase orders. Table will have only 3 fields, client, company code and range. Before saving data on maintenance view, we will check both company code and range and if they are not valid we will warn user and will not let data to be saved.

Table

Dictionary: Display Table

Transparent Table: **ZMM_T_PO_NR_CUST** Active
Short Description: PO Custom Number Range(EINKBELEG) Per Company Code SE11 View

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes

Field	Key	Init...	Data element	Data Type	Length	Decim...	Coordinate	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0		0 Client
BUKRS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	BUKRS	CHAR	4	0		0 Company Code
RANGE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NRNR	CHAR	2	0		0 Number range number

Table Structure

Utilities

- Settings...
- Display Object List Ctrl+Shift+F5
- Worklist
- Add to Pool of Inactive Objects
- Display Navigation Window Ctrl+Shift+F4
- Activation Log
- Database Object
- Runtime Object
- Assign Authorization Group
- Table Maintenance Generator**
- Table Contents
- Where-Used List Ctrl+Shift+F3
- Versions

Dictionary: Display Table

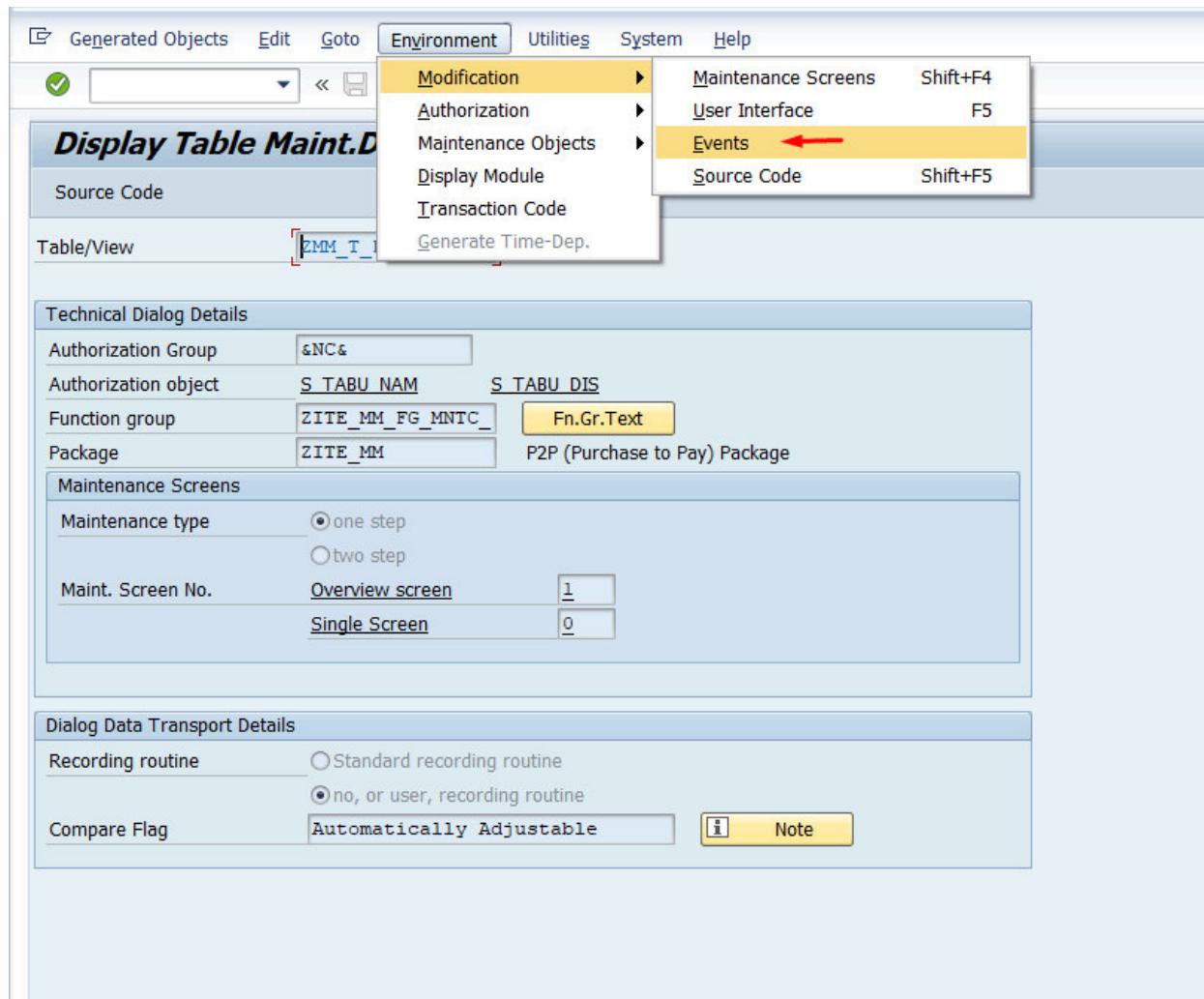
Transparent Table: **ZMM_T_PO_NR_CUST** Active
Short Description: PO Custom Number Range(EINKBELEG) Per Company Code SE11 View

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes

Field	Key	Init...	Data element	Data Type	Length	Decim...	Coordinate	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0		0 Client
BUKRS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	BUKRS	CHAR	4	0		0 Company Code
RANGE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NRNR	CHAR	2	0		0 Number range number

Open Maintenance View

I assume you have created a maintenance view. And willing to add event to validate data. If you don't know how to create maintenance view, please [read on that link](#).



Add Event to Maintenance View

Add event before save and give a form name to be called when event is triggered.

The screenshot shows the SAP SM30 maintenance view interface. At the top, there is a menu bar with options like Table View, Edit, Goto, Selection, Utilities, System, and Help. Below the menu is a toolbar with various icons. The main title of the view is "Display View "FORM routines to be called from view maintenance"". Underneath the title, there are four small icons: a green checkmark, a magnifying glass, a document, and an information sign. The "View/table" field contains the text "ZMM_T_PO_NR_CUST". A red box highlights this text, and a red arrow points to it with the label "01- ADD Event Before Save". Below this, a table titled "FORM routines to be called from view maintenance" is displayed. The table has three columns: T (Type), FORM routine, and Editor. One row is visible, showing "01 BEFORE_SAVE" in the FORM routine column and an editor icon in the Editor column. A red arrow points to the editor icon with the label "02- Open Editor".

Add Event and Form Name

Code Block

Click open editor and create form that you have given above (BEFORE_SAVE in that example)

```

1  * -----
2  *** INCLUDE LZITE_MM_MM_MNTE_F01.
3  *
4
5  FORM before_save.          Create Form routine with same name you have given
6                                     in previous screen
7
8  DATA : recs TYPE TABLE OF zmm_t_po_nr_cust,
9        rec  TYPE zmm_t_po_nr_cust.
10
11 LOOP AT total.
12   IF <vim_total_struct> IS ASSIGNED AND ( <action> EQ 'N' OR <action> EQ 'U' ). New
13     CLEAR rec.
14     MOVE-CORRESPONDING <vim_total_struct> TO rec.
15     APPEND rec TO recs.
16     ENDIF.                                Update
17   ENDLOOP.
18
19   "Validate changes
20   IF recs IS NOT INITIAL.
21     DATA(status) = zmm_cl_po_numbering->validate_new_entries( recs = recs ). To abort saving
22     IF status-status NE zutils_cl_defs->c_stat_success.
23       vim_abort_saving = abap_true.
24       MESSAGE status-status_text TYPE 'S' DISPLAY LIKE 'E'.
25     ENDIF.
26   ENDIF.
27
28 ENDFORM.

```

FORM Content of BEFORE_SAVE

Above, from line 9 to 15 are used to collect new records or changed records data to recs internal table. **total** table contains all records and when we loop on total records, **<vim_total_struct>** field-symbol is assigned, by checking **<action>** field-symbol we can learn about state of record, if it is new or updated or being deleted.

Later on line 19, recs table is passed to a class that contains validation logic. That class method returns a status structure. If result is not success. we are setting **vim_abort_saving = abap_true** on line 21. In that way, saving will not happen.

And we are showing message to inform user on line 22.

Here are the lines, that will be useful for you.

```
FORM before_save.  
  "Collect records to be checked  
  DATA : recs TYPE TABLE OF zmm_t_po_nr_cust,  
         rec  TYPE zmm_t_po_nr_cust.  
  LOOP AT total.  
    IF <vim_total_struct> IS ASSIGNED AND ( <action> EQ 'N' OR <action>  
      CLEAR rec.  
      MOVE-CORRESPONDING <vim_total_struct> TO rec.  
      APPEND rec TO recs.  
    ENDIF.  
  ENDLOOP.  
  
  "Validate changes  
  IF recs IS NOT INITIAL.  
    "Place your logic below  
    DATA(status) = zmm_cl_po_numbering=>validate_new_entries( recs =  
    IF status-status NE zutils_cl_defs=>c_stat_success.  
      vim_abort_saving = abap_true. "To abort saving  
      MESSAGE status-status_text TYPE 'S' DISPLAY LIKE 'E'.  
    ENDIF.  
  ENDIF.  
  
ENDFORM.
```

Activate your changes and go to sm30 to test.

The screenshot shows the SAP SM30 maintenance view titled "New Entries: Overview of Added Entries". The table displays a single row with columns "Company" and "Range". The "Company" column contains "XX" and the "Range" column contains "ZZ". The cell for "Company" is highlighted with a yellow background and has a red border around its top-left corner, indicating a validation error. A small magnifying glass icon is positioned at the bottom right of the "Range" cell. Below the table, a message bar shows an error: "XX Invalid company code!". The SAP logo and the menu "SM30" are visible at the bottom.

Try to save data with invalid company code

That is all. I hope that is useful for you.

Thanks for reading.

Reference Links

<https://blogs.sap.com/2015/10/29/validate-data-in-table-maintenance-generator-event/>

<https://blogs.sap.com/2012/03/12/viewmaintain-subset-of-data-in-table-maintenance-generator/>

Tags:

sm30table

Comments

R

rkrish18 Explorer

2023 Sep 26
12:56 AM

Great Blog. I just have one question: there is any way to allow the user to correct the entry? The validation works in stopping the user, but it greyed out the field, so the user is not able to correct the entry and it has to leave SM30 completely and get back in again. I tried using just a success message, but the result was the same.

Best Regards:

Luis Lara

M

mgross1 Participant

2024 May 24
1:38 PM

-

edited

2024 May 24
1:39 PM

- edited

Why not just use a foreign key to the company code master? Then there will be automatic validation in SM30.

